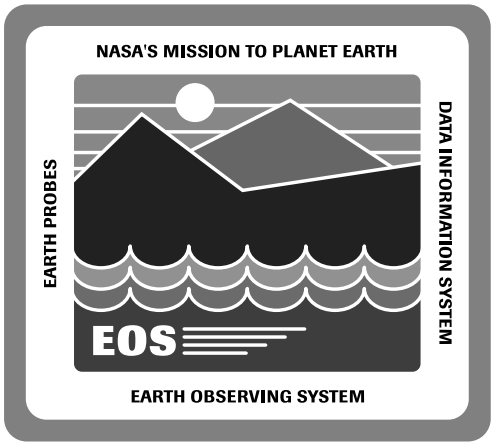


Communications Subsystem (CSS) Overview

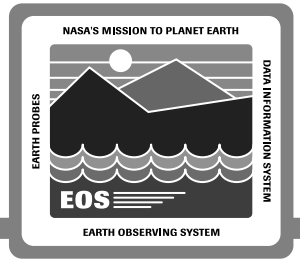
Evan Winston

18 January 1995

CSMS PDR Agenda - Day 2

Communications Subsystem (CSS)	**08:00-10:30**
Break	**10:30-10:45**
Communications Subsystem (Continued)	**10:45-12:00**
Lunch	**12:00-13:00**
CSMS EP4 Briefing/Demo	**13:00-15:00**
Daily Wrap-Up	**15:00-16:00**

CSS Roadmap



CSS Overview

Services/Interfaces
Client/Server Development
Message Passing

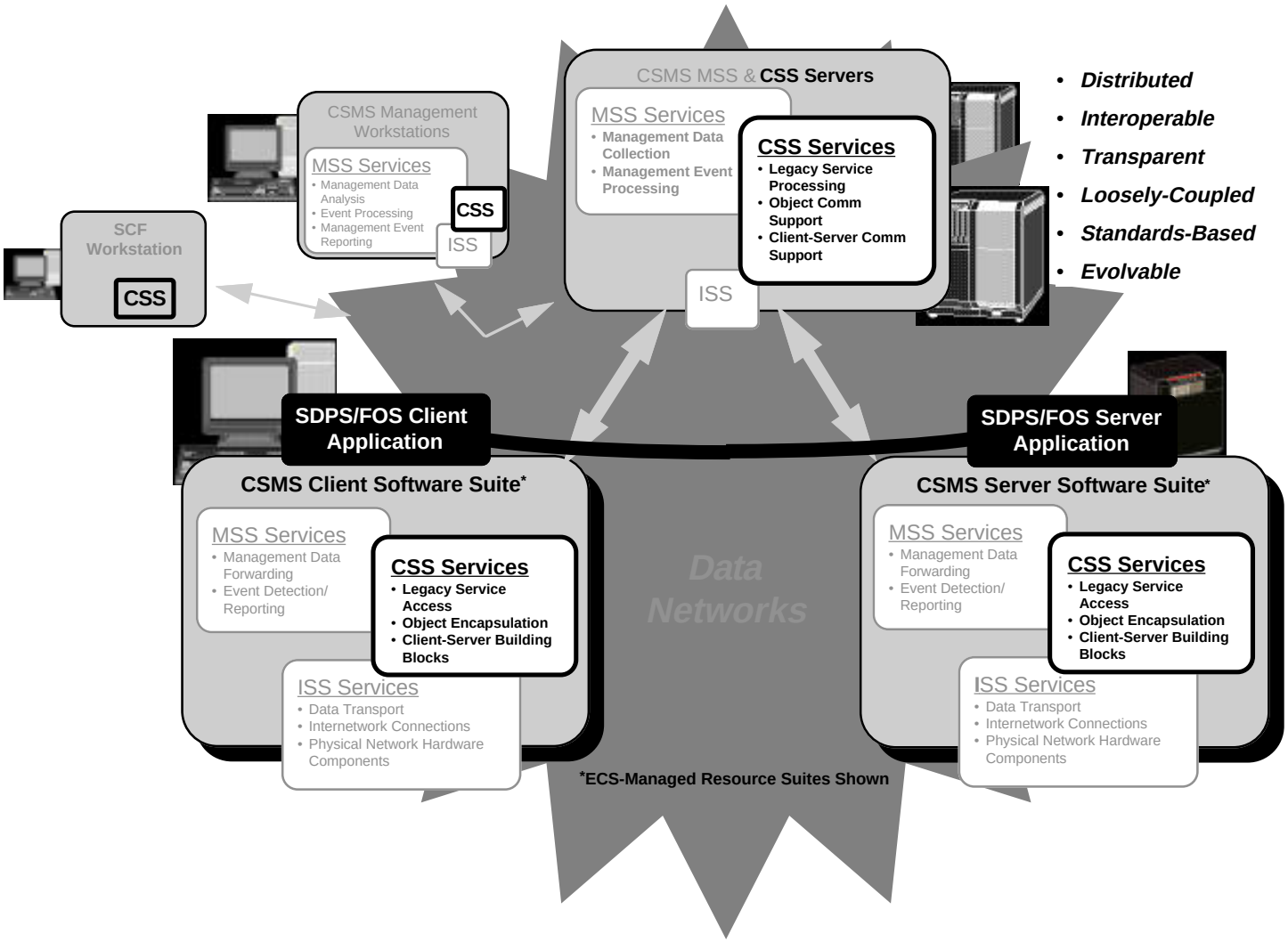
Break

Client/Server Event Flow
Cell Topology

Technology Migration

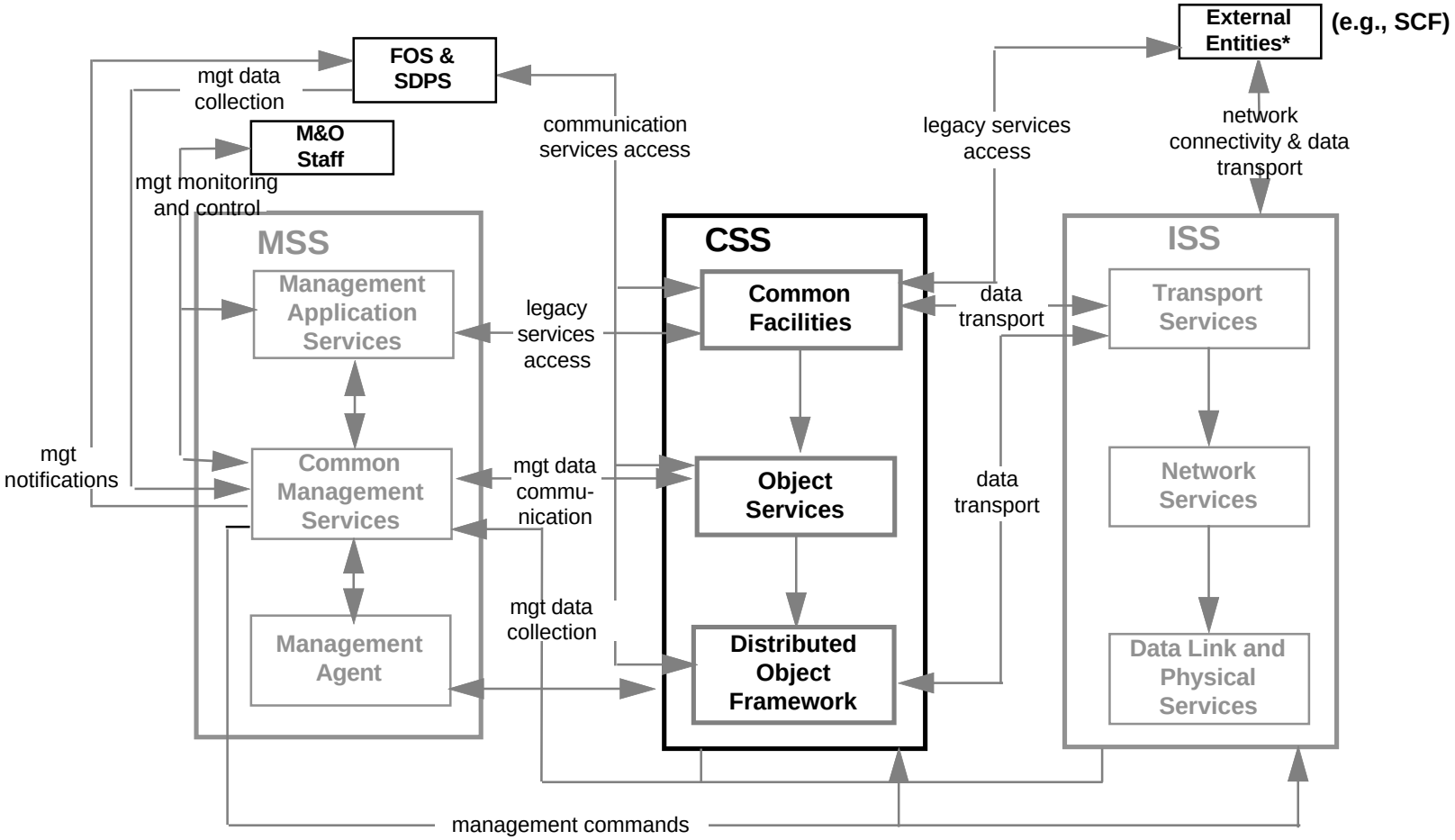
Wrap-up

CSMS Physical Context



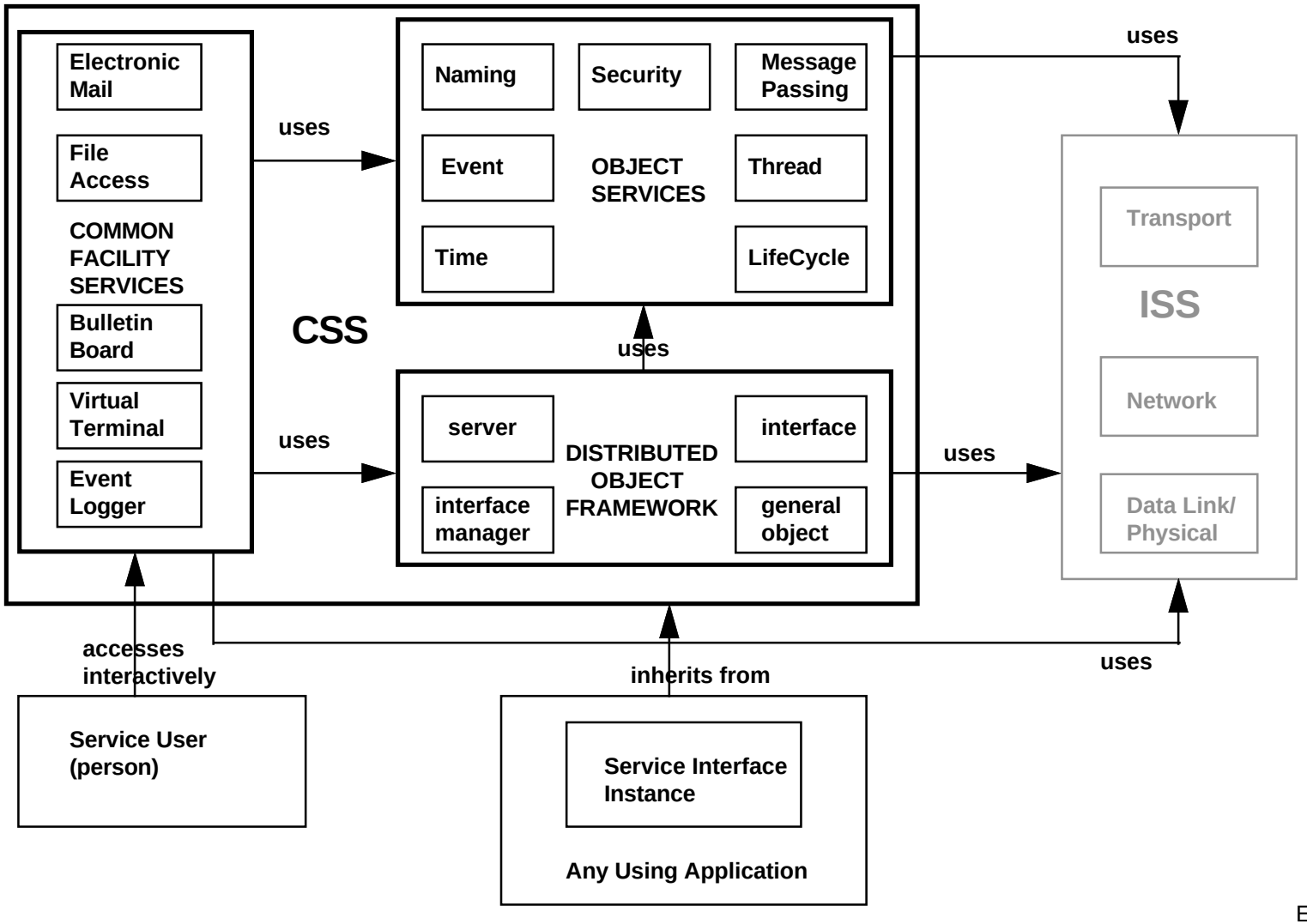
- ***Distributed***
- ***Interoperable***
- ***Transparent***
- ***Loosely-Coupled***
- ***Standards-Based***
- ***Evolvable***

CSMS Subsystem Overview and Primary Service Flows

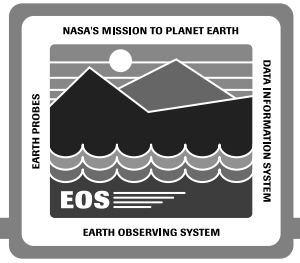


*A subset of CSS services is included in ECS Toolkits

CSS Subsystem Design



Progress Since SDR



CSS Service Class

ORB Services → **Distributed Object Framework (DOF)**

OMG CORBA 2.0 Services

Release B → **Release C**

DCE Encapsulation Method

Custom Encapsulation → **OODCE + Custom Wrapper**

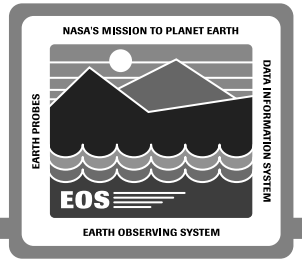
Service Usage Mapping

		CSS Object Services					CSS Common Facilities				CSS DOF
Segment	**Subsystem**	**Sec**	**Dir**	**Thread**	**Evnt**	**Msg**	**File**	**Virtual Terminal**	**email**	**BBS**	**DOF**
FOS	Command					●			●		●
	Resource Management					●			●		●
	User Interface					●			●	●	●
	Data Management					●					●
	Analysis				●	●			●		●
	Telemetry				●	●			●		●
	Planning & Scheduling					●					●
	Command Management					●			●		●
SDPS	Client	●	●				●	●	●	●	●
	Interoperability	●	●	●	●	●	●	●	●	●	●
	Data Management						●			●	●
	Data Server			●			●		●		●
	Data Processing					●	●		●	●	●
	Planning					●	●		●	●	●
	Ingest					●	●		●		●
MSS	ALL	●	●		●	●	●		●	●	●

Characterization of Service Requirements by Release

Service Grouping	Major Service Requirement	IR1 Services	Release A Services
Distributed Object Framework	Interoperability framework for OO client-server development & execution	Developmental framework (class libraries, language bindings, RPC, sockets)	Execution framework for distributed object implementation
Object Services	Interprocess communication and specialized infrastructural services	Core services via object interfaces (DCE security, time, CDS directory/ naming)	Additional services via object interfaces (event, logging, message passing, threads)
Common Facilities	Legacy services for file access/ transfer, e-mail, bulletin board and remote terminal support	Initial legacy services (ftp, telnet)	Additional legacy services with applications & security I/Fs (kftp, ktelnet, e-mail, BBS, DFS)

Summary of PDR Trades and Analyses



Analysis	Objective	Conclusion	Future Analysis
DCE Cell Configuration	Evaluate several cell configurations and their applicability to the needs of ECS.	Cell-per-DAAC configuration along with an external Isolation Cell (ISO+) is recommended. Cell-per-DAAC provides better security, autonomy of DAACs, and scalability. The ISO+ on top of the internal cell-per-DAAC provides a clear separation of external ECS users from internal ECS users providing better security. Implementing policy on these external users will become easier and more uniform than without an ISO+.	Since cell configuration of the entire ECS is very important and can affect all the elements in ECS, CSMS plans to consult with other large DCE deployers for their lessons learned and insights in this area.
Message Passing	Evaluate implementation options that satisfies the FOS and SDPS requirements for asynchronous and deferred synchronous message passing.	The recommendation is to select a suitable COTS product that supports asynchronous message passing and custom implement deferred synchronous message passing, and other desirable features like security and callbacks.	Continue proof-of-concept prototyping with FOS.
Multicast Analysis	The main objectives of this multicast analysis are to: 1. Determine different methods of multicasting telemetry data on the FOS network. 2. Identify the potential for prototyping the solutions discovered in item (1). 3. Analyze the impact of cost and policy issues pertaining to the proposed solutions.	CSMS recommends implementing multicast by using Reliable Multicast Protocol (RMP) to provide reliable communications and C++ interfaces to the developers. CSMS also recommends using MBONE technology and multicast routers that provide direct multicast support	On the basis of this investigation and keeping in mind the PDR schedule, CSMS further recommends that proof-of-concept prototyping be initiated after the PDR season.

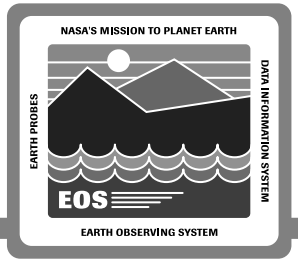
Summary of PDR Trades and Analyses (cont.)

Analysis	Objective	Conclusion	Future Analysis
DCE Encapsulation	Determine method for DCE encapsulation.	CSS recommends OODCE as the DCE Encapsulation Technique through Release B.	Track maturation of CORBA 2.0 products. Continue prototyping of CORBA products.
Remote File Access (RFA)	Analyze requirements for RFA, evaluate available alternatives (COTS) that provide the functionality and recommend implementation options	DFS ranks highest in the evaluation. It must be noted that DFS currently does not have all the capabilities that the Data Server Group needs. Efforts are underway in a DCE SIG to include these capabilities in the standard DFS.	Continue joint proof-of-concept prototyping effort with the Data Server Group using AFS and MR-AFS.

Release A Major COTS/Custom Choices, by CSMS Subsystem

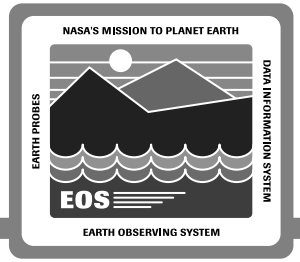
COTS/Custom	Product or Product Class	Purpose
COTS	OODCE, DCE	Developmental framework (class libraries, language binding, RPC, sockets) and execution framework for distributed object implementation; development services via object interfaces (security, time, naming, event, logging, message passing, threads, DFS)
COTS	ftp, kftp, telnet, ktelnet, e-mail, BBS, X Window http / WWW (not in A baseline)	IP-based legacy services for file transfer, e-mail, bulletin board and remote terminal support
Custom	Encapsulation Code	Custom code to provide developer interfaces to OODCE and DCE services; custom code to customize (e.g., API support) legacy services for ECS developers.

Components of CSMS Security Implementation



Security Need	CSMS Security Implementation	CSS
Authentication	DCE-based Kerberos, Kerberized ftp, Kerberized telnet	X
Authorization and access control	DCE access control; Router-based filters (port/socket at transport layer, and source and/or destination address at network layer); DCE cell configuration / "isolation-cell" partitioning	X
Data integrity	DCE-based RPCs (encrypted checksums)	X
Data confidentiality	DCE-based RPCs (encrypted data) used as required	X
Counter measures for degradation in network or processing resource performance through denial of service attack	Router-based filters	
Security database management	DCE ACL managers, registry database	
Compliance management	MSS COTS & public domain tools for password audits, file system integrity checking	
Intrusion detection	COTS for detecting viruses, worms, Trojan horses, public domain tools (e.g, TCP Wrapper)	
Security reporting	RDBMS	

CSS Security



Based on Kerberos Version 5

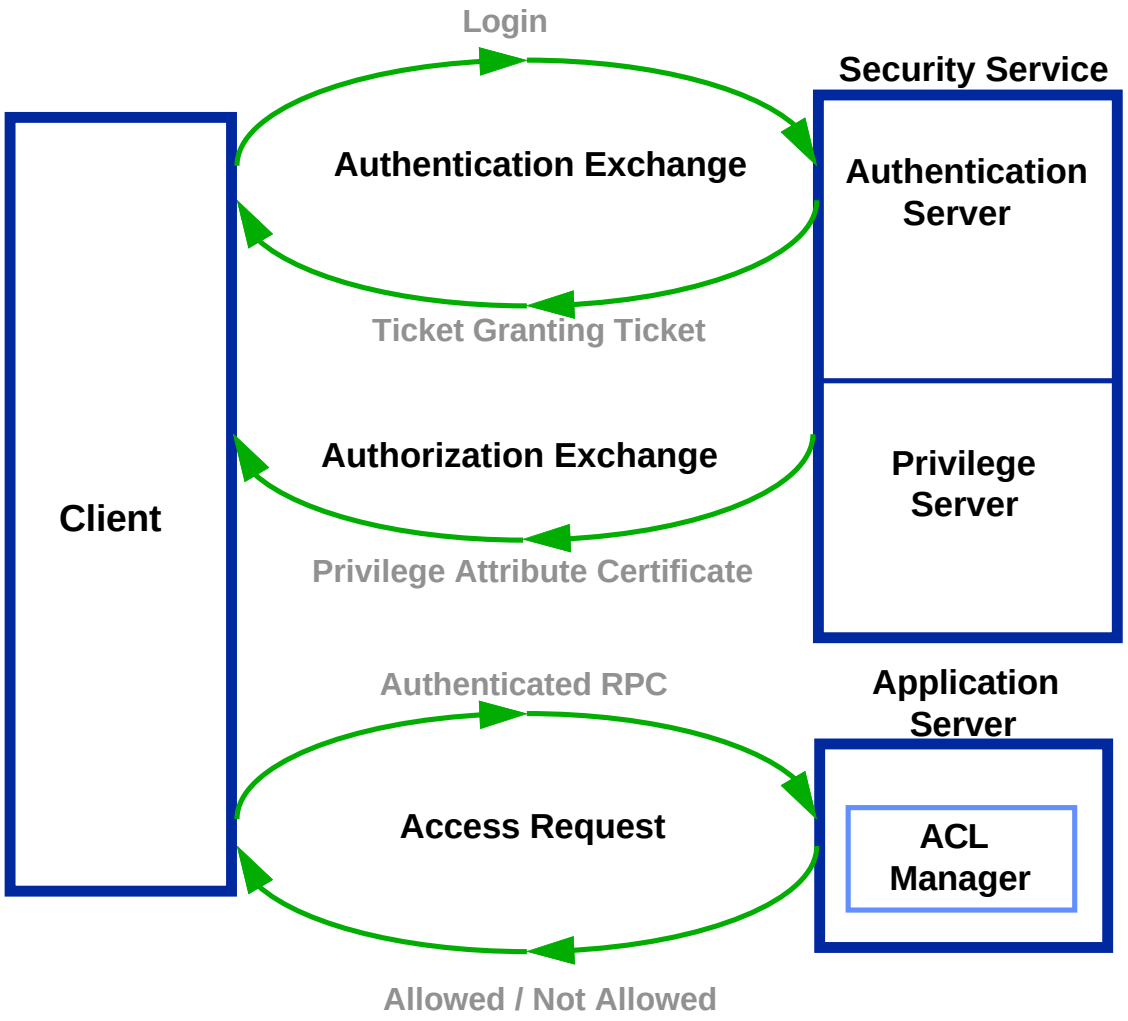
Features

- **Authentication**
- **Authorization**
- **Data Integrity**
- **Data Confidentiality**

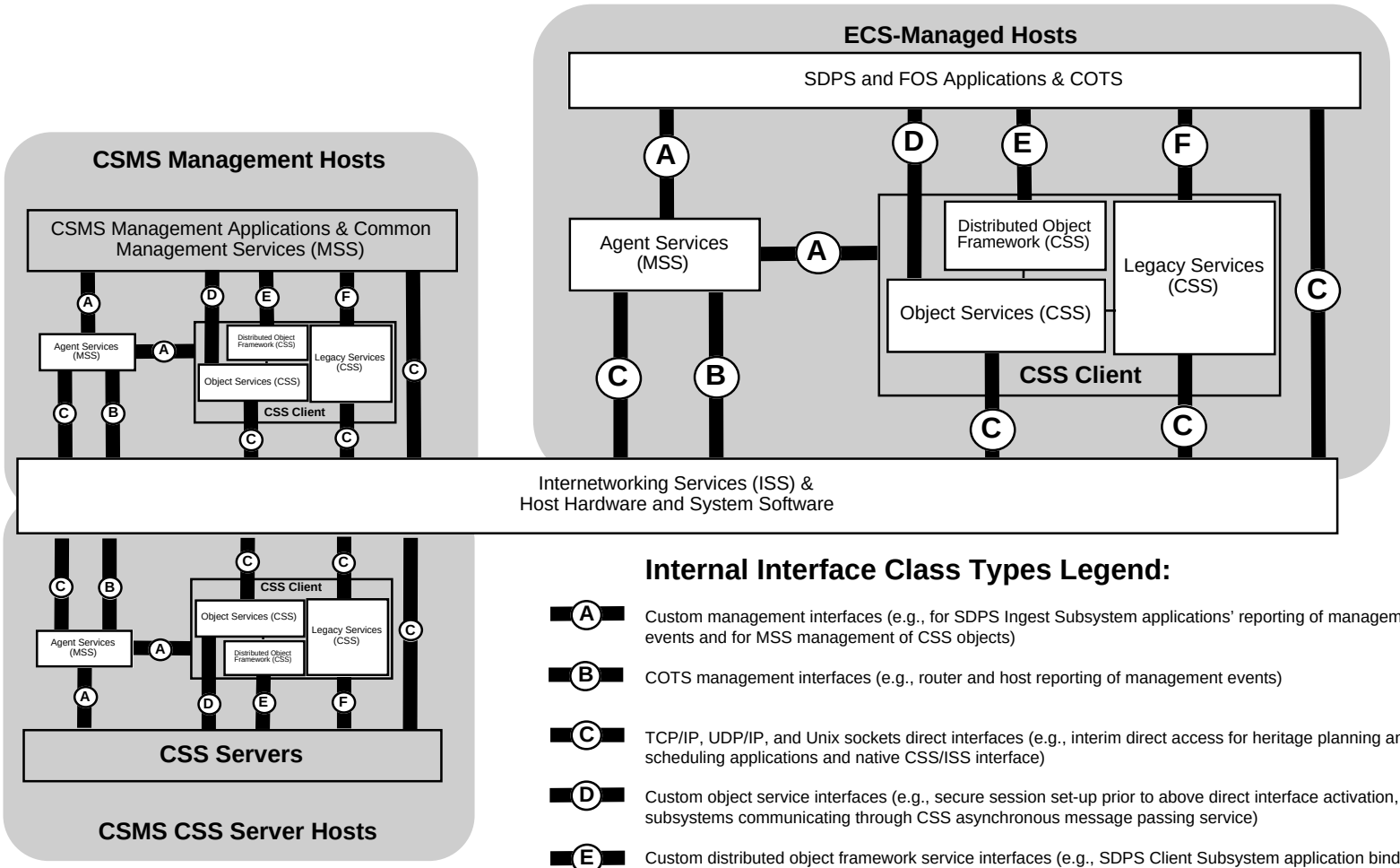
Benefits

- **Fully integrated with DCE**
- **Provides migration to public key**
- **Kerberos developed by MIT**
- **Standards based / COTS based**
- **Compatibility with Public Domain Kerberos**
- **Well tested**
- **No Password in the clear**

Security Concept



CSMS Internal Interfaces



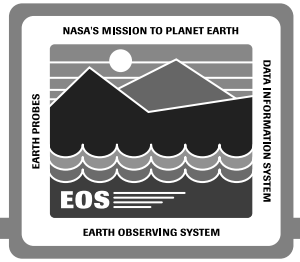
Internal Interface Class Types Legend:

- **A** — Custom management interfaces (e.g., for SDPS Ingest Subsystem applications' reporting of management events and for MSS management of CSS objects)
- **B** — COTS management interfaces (e.g., router and host reporting of management events)
- **C** — TCP/IP, UDP/IP, and Unix sockets direct interfaces (e.g., interim direct access for heritage planning and scheduling applications and native CSS/ISS interface)
- **D** — Custom object service interfaces (e.g., secure session set-up prior to above direct interface activation, FOS subsystems communicating through CSS asynchronous message passing service)
- **E** — Custom distributed object framework service interfaces (e.g., SDPS Client Subsystem application binding to a particular Data Server Subsystem server)
- Custom legacy service (CSS) interfaces (e.g., product delivery to scientist from SDPS Data Server application through ftp)

CSS Internal Interface Summary

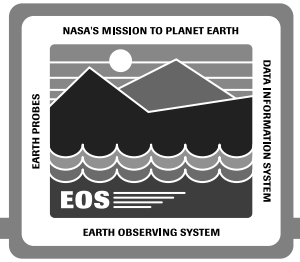
Interface Category	Interface	Purpose	Provided by/via	Used by	Notes/Examples
E	Distributed Object Framework	Provides an encapsulation service to Object Services and to ease Client-Server development in a distributed environment	CSS/DOF	Required to be used by all ECS client and server applications	Any ECS client and server access CSS services through this framework. Only specialized applications will need to access Object Services directly through interface type D
F	Send Email Message	Allows any application to send an electronic mail message to a designated recipient(s)	CSS / Email API	Optional use by any ECS application	May be used to notify an SCF QA inspector that a given product is available, notify a user that a media product has been shipped, etc. Undeliverable messages will be logged for M&O handling
F	Send BBS Message	Allows any ECS application to send material to the ECS BBS	CSS-BB API	Optional use by any ECS application	May be used to notify a given interest group of a new service, the planned suspension of a service, service outage, to update a user registration form, etc. Once posted, material will be maintained/deleted by M&O
F	Secure ftp	Allows any ECS application to exchange files with remote toolkit users	CSS/File Transfer API and Command Line Interface	Required by all secure applications in ECS (SDPS, FOS and CSMS)	To be used to transfer files between non-registered ECS users and secure ECS hosts
F	Secure Telnet	Allows any ECS applications to establish secure sessions with ECS hosts	CSS/Command Line Interface	Required to be used by all secure sessions in ECS (SDPS, FOS and CSMS)	To be used for command-line access between non-registered ECS users to access secure ECS hosts

CSS Internal Interface Summary (cont.)



Interface Category	Interface	Purpose	Provided by/via	Used by	Notes/Examples
D	Directory Service Interface	To register special attributes and special interfaces to Directory Services	CSS/Directory Services API and Classes	Optional use by any ECS application to store/ retrieve special attributes into directory/naming	FOS requested use of special attributes
D	Security Services Interface	Allows ECS applications to authorize clients to access secure resources	CSS/Security Service API and Classes	Required use by every secure ECS application to provide ACL based authorization to restrict access to resources	ACL Manager use by application servers
D	Threads Services Interface	Allows ECS applications to use threads	CSS/COTS API	Optional use by any ECS application to write multi-threaded applications	For use by SDPS and FOS applications especially when they have multiple clients accessing the same server interface
D	Object Services Interface	Allows ECS applications to use Distributed Time Service	CSS/COTS API	Optional use by any ECS application to interface the time service	Direct interface to time service is still through the native Operating Sys. time service API
D	Lifecycle Services Interface	Initialization and activation of objects and applications	CSS/Life Cycle API and classes	Optional use by any ECS application to initialize and activate object and applications on demand	
D	Event Services Interface	Asynchronous communication	CSS/Event Services Classes and API	Optional use by certain SDPS applications	
D	Message Passing Interface	Asynchronous and deferred coupled synchronous communications	CSS/Message Passing API	Optional use by any ECS application to send messages to designated receiver(s)	Primarily used for point to point communications

Mitigating the Risks



Summary:

Survey vendors and emerging standards aggressively

Prototype, prototype, prototype

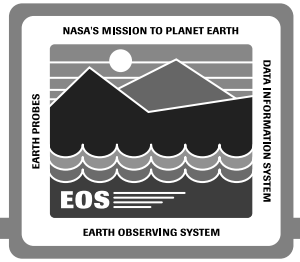
Participate actively in standards groups, users groups, and consortia

Design top down and bottom up

Plan ahead for technology migrations

Always have fallback positions and decision points

CSS Roadmap



CSS Overview

Services/Interfaces

Client/Server Development

Message Passing

Break

Client/Server Event Flow

Cell Topology

Technology Migration

Wrap-up